PAPER P-1135

A TARGETING MODEL THAT MINIMIZES COLLATERAL DAMAGE

Jeffrey H. Grotte

December 1975

INSTITUTE FOR DEFENSE ANALYSES
PROGRAM ANALYSIS DIVISION

IDA Log No. HQ 75-17630
Copy 61 of 125 copies

The work reported in this document was conducted under Contract DAHC15 73 C 0200 for the Weapons Systems Evaluation Group. The publication of this IDA Paper does not indicate endorsement by the Weapons Systems Evaluation Group, nor should the contents be construed as reflecting the official position of that agency.

PAPER P-1135

A TARGETING MODEL THAT MINIMIZES COLLATERAL DAMAGE

Jeffrey H. Grotte

December 1975

INSTITUTE FOR DEFENSE ANALYSES
PROGRAM ANALYSIS DIVISION
400 Army-Navy Drive, Arlington, Virginia 22202

Contract DAHC15 73 C 0200
Task T-237

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

Field	Entry
1. REPORT NUMBER	P-1135
2. GOVT ACCESSION NO.	
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	A Targeting Model That Minimizes Collateral Damage
5. TYPE OF REPORT & PERIOD COVERED	Final
6. PERFORMING ORG. REPORT NUMBER	P-1135
7. AUTHOR(s)	Jeffrey H. Grotte
8. CONTRACT OR GRANT NUMBER(s)	DAHC15 73 C 0200
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Institute for Defense Analyses Program Analysis Division 400 Army-Navy Drive, Arlington, Virginia 22202
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	T-237
11. CONTROLLING OFFICE NAME AND ADDRESS	Weapons Systems Evaluation Group 400 Army-Navy Drive, Arlington, Virginia 22202
12. REPORT DATE	December 1975
13. NUMBER OF PAGES	32
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	
15. SECURITY CLASS. (of this report)	Unclassified
15a. DECLASSIFICATION DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)	Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)	Collateral Damage, Targeting Model, Lexicographic Enumeration, Nonconvex Programming, Integer Programming, Lawler-Bell Algorithm

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper considers the problem of allocating weapons to achieve targeting objectives while simultaneously minimizing the aggregate damage to surrounding nonmilitary facilities, each of which has an upper limit to the damage it is permitted to incur. This problem is, in general, nonconvex. A model is formulated that assumes only that damage to individual targets or associated nonmilitary facilities does not decrease as the number of allocated weapons increases. An implicit

20. continued

enumeration algorithm, based on that of Lawler and Bell, is described that yields optimal integer solutions. An example is presented.

CONTENTS

PREFACE

This paper considers the problem of allocating weapons to achieve targeting objectives while simultaneously minimizing aggregate damage to surrounding nonmilitary facilities, each of which has an upper limit to the damage it is permitted to incur. A model is formulated which assumes only that damage to individual targets or associated facilities does not decrease as the number of allocated weapons increases. An implicit enumeration algorithm, based on that of Lawler and Bell (see Reference [3]), is described that yields optimal integer solutions. An example is presented.

This paper differs from IDA paper P-1106 (Reference [2]) in that it presents the full generality of the collateral damage minimizing model, whereas P-1106 describes a model (NDM) tailored to specific design requirements. In addition, the code listed in the Appendix may prove a prototype for a modified NDM with greatly decreased run times.

A TARGETING MODEL THAT MINIMIZES COLLATERAL DAMAGE

One of the assumptions behind the argument to employ *counterforce* targeting of strategic weapons (the targeting of an enemy's strategic capability), as opposed to *countervalue* targeting (the objective of which is the destruction of population and economy), is that sufficient destruction of strategic targets can be achieved without causing appreciable damage to the surrounding nonstrategic facilities. This paper presents a model which addresses the following two questions: Given a collection of weapons, potential aimpoints, and a configuration of strategic targets--each being assigned a *minimum* level of damage; and nonstrategic facilities--each having a *maximum* level of permissible damage,

(A) Is there an assignment of weapons to aimpoints (an *allocation*) that satisfies the above two sets of constraints?

(B) Of all allocations satisfying the above two sets of constraints, what is the one (or *a* one) that minimizes the (perhaps weighted) sum of the damage to the nonstrategic facilities?

I. MATHEMATICAL FORMULATION

The fundamental elements of the model are M strategic targets, henceforth called simply "targets," N nonstrategic facilities, or "nontargets," I different weapon types, and J potential aimpoints to which any weapon can be directed. An allocation $\underset{\sim}{z}$ is the matrix $\{z_{i,j} | i=1,\ldots,I;\ j=1,\ldots,J\}$ where $z_{i,j}$, an integer, is the number of weapons of type i allocated to aimpoint j.

For each target m, we suppose a real-valued response function $f_m(\underset{\sim}{z})$ which represents the damage to target m from allocation $\underset{\sim}{z}$. We require that $f_m(\underset{\sim}{z})$ be monotonically nondecreasing in each component of $\underset{\sim}{z}$, which is an implicit assumption that, given any allocation, the allocation of additional weapons does not result in less damage to any target. Each target m is assigned a real number, c_m, which is the minimum damage requirement (targeting objective), i.e., for an allocation $\underset{\sim}{z}$ to be feasible, it must satisfy $f_m(\underset{\sim}{z}) \geq c_m$, m=1,...,M.

Similarly, for each nontarget n there is a response function $g_n(\underset{\sim}{z})$, monotonically nondecreasing in each component of $\underset{\sim}{z}$, and a real number d_n denoting the maximum damage permitted to this nontarget. Further, each nontarget n is assigned a nonnegative weight, or value, λ_n.

The nonnegative integer w_i is the number of weapons of type i available for allocation.

We can now combine questions (A) and (B) into the following problem P:

$$\underset{\underset{\sim}{z}}{\text{Minimize}}\ h(\underset{\sim}{z}) \equiv \sum_{n=1}^{N} \lambda_n g_n(\underset{\sim}{z}) \qquad \text{subject to} \tag{1}$$

$$f_m(\underset{\sim}{z}) \geq c_m \qquad m=1,\ldots,M\ ; \tag{2}$$

$$g_n(\underset{\sim}{z}) \leq d_n \qquad n=1,\ldots,N\ ; \tag{3}$$

$$\sum_{j=1}^{J} z_{i,j} \leq w_i \qquad i=1,\ldots,I\ ; \tag{4}$$

$$z_{i,j} \in \mathbb{Z}^+ \qquad i=1,\ldots,I;\ j=1,\ldots,J\ ; \tag{5}$$

where $\mathbb{Z}^+$ is the set of nonnegative integers. If problem P is infeasible, then the answer to question (A) is clearly "no," otherwise an answer to question (B) is ensured because the number of allocations which satisfy constraints (4) and (5) is finite.

II. AN ALGORITHM

Problem P admits solution by implicit enumeration. The following algorithm is based upon the lexicographic technique of Lawler and Bell (see Reference [3])--though, unlike the Lawler-Bell approach, this algorithm does not use binary vectors. We first identify the matrix $\underset{\sim}{z}$ with a vector $\hat{\underset{\sim}{z}}$. This can be done in a number of ways, one of which is through the following relationship:

$$\hat{z}_k = z_{i,j} \qquad k=i + (j-1)\cdot I; \; i=1,\ldots,I; \; j=1,\ldots,J \; . \tag{6}$$

Note that this can be reversed as follows:

$$z_{i,j} = \hat{z}_k, \; i = k - \left\langle \frac{k-1}{I} \right\rangle \cdot I, \quad j = \left\langle \frac{k-1}{I} \right\rangle + 1; \quad k=1,\ldots,K=I\cdot J.$$

where $\langle x \rangle$ is the largest integer less than or equal to x. With this in mind, we will drop the circumflex, and in the discussion that follows, all allocations will be vectors in $\mathbb{Z}_K^+$, i.e., K-dimensional vectors with nonnegative integer components. We require two binary relations between vectors in $\mathbb{Z}_K^+$:

Componentwise (partial) Ordering:

We write $\underset{\sim}{x} \geq \underset{\sim}{y}$ if $x_k \geq y_k \qquad k=1,\ldots,K$

$\underset{\sim}{x} > \underset{\sim}{y}$ if $\underset{\sim}{x} \geq \underset{\sim}{y}$ and $x_k > y_k$ for at least one k.

Lexicographic Ordering:

We write $\underset{\sim}{x} \underset{L}{>} \underset{\sim}{y}$ if $x_{k'} > y_{k'}$ where $k' = \max_{1 \leq k \leq K} \{k | x_k \neq y_k\}$,

and $\underset{\sim}{x} \underset{L}{\geq} \underset{\sim}{y}$ if $\underset{\sim}{x} \underset{L}{>} \underset{\sim}{y}$ or $\underset{\sim}{x} = \underset{\sim}{y}$.

Let $\mathcal{S} = \left\{ \underset{\sim}{z} \in \mathbb{Z}_K^+ | z_k \leq w_h \text{ for } k=1,\ldots,K, \; h=k-\left\langle \frac{k-1}{I} \right\rangle \cdot I \right\}$.

Thus $\mathcal{S}$ is a set of allocations that satisfy constraint (5) of problem P, and clearly contains all allocations that satisfy constraint (4), and so must contain all solutions to problem P providing problem P is feasible. Since $\underset{L}{\geq}$ totally orders $\mathcal{S}$, we could enumerate all the points of $\mathcal{S}$ and find the solution to P in this manner. However, the monotonicity of the

objective and constraint functions will permit us to skip over many infeasible and/or nonoptimal points. To see this, we need some notation. Consider a vector $\mathbf{z} \epsilon \mathcal{S}$. We will denote by $\mathbf{z}+1$ the vector $\mathbf{x}$, if it exists, satisfying

$$\begin{cases} \mathbf{x} \ \epsilon \ \mathcal{S} \\ \mathbf{x} \underset{L}{>} \mathbf{z} \\ \mathbf{y} \underset{L}{>} \mathbf{z} \Rightarrow \mathbf{y} \underset{L}{>} \mathbf{x} \ . \end{cases} \tag{7}$$

At most one such vector exists, but may fail to exist because of the boundedness of $\mathcal{S}$. The vector $\mathbf{z}-1$ will be that vector $\mathbf{x}$, if it exists, satisfying

$$\begin{cases} \mathbf{x} \ \epsilon \ \mathcal{S} \\ \mathbf{z} \underset{L}{>} \mathbf{x} \\ \mathbf{z} \underset{L}{>} \mathbf{y} \Rightarrow \mathbf{x} \underset{L}{>} \mathbf{y} \ . \end{cases} \tag{8}$$

This vector will always exist except for $\mathbf{z} \equiv 0$. The vector $\mathbf{z}^*$ will be that $\mathbf{x}$, if it exists, satisfying

$$\begin{cases} \mathbf{x} \ \epsilon \ \mathcal{S} \\ \mathbf{x} \underset{L}{>} \mathbf{z} \\ \mathbf{x} \ngeq \mathbf{z} \\ \left(\left(\mathbf{y} \underset{L}{>} \mathbf{z}\right) \wedge \left(\mathbf{y} \ngeq \mathbf{z}\right) \right) \Rightarrow \mathbf{y} \underset{L}{\geq} \mathbf{x} \ . \end{cases} \tag{9}$$

Intuitively, $\mathbf{z}^*$ is the first vector in $\mathcal{S}$ following $\mathbf{z}$ (in the lexicographic ordering) which is not (componentwise) greater than or equal to $\mathbf{z}$. For some $\mathbf{z}$, $\mathbf{z}^*$ may not exist; however, we will adopt the following convention: For any $\mathbf{z}$ for which $\mathbf{z}^*$ does not exist, we will set

$$(\mathbf{z}^*-1)_k = w_h \quad \text{for } h=k-\left\langle \frac{k-1}{I} \right\rangle \cdot I, \ k=1,\ldots,K ,$$

thereby ensuring that $\mathbf{z}^*-1$ exists for every $\mathbf{z} \epsilon \mathcal{S}$. Crucial to the algorithm is the observation that for any $\mathbf{z} \epsilon \mathcal{S}$, any $\mathbf{y}$ that satisfies $\mathbf{z} \underset{L}{\leq} \mathbf{y} \underset{L}{\leq} \mathbf{z}^*-1$ also satisfies $\mathbf{y} \geq \mathbf{z}$.

Figure 1 outlines the fundamentals of the algorithm. A brief inspection of the flow chart will make clear that the algorithm must terminate after a finite number of steps. If $\bar{h} = \infty$ upon termination, the problem is infeasible, otherwise an optimum integer allocation will always be found. The order in which the constraints are examined was chosen because, for certain applications, this order was efficient. However, we make no claim that this is, in any sense, an optimal ordering. For other applications, a different sequence of constraint evaluations might well prove to be better.

III. A CLASS OF EXAMPLES

We will now look at a class of examples with point targets and nontargets, where the destruction of any target or nontarget is a binomial random variable with probability of kill dependent on the allocation, but with independent weapons effects. We will use Cartesian coordinates to specify location, in particular, target coordinates are (x_m, y_m), $m=1,\ldots,M$; nontarget coordinates are (μ_n, ν_n), $n=1,\ldots,N$; and aimpoint coordinates are (ξ_j, ζ_j), $j=1,\ldots,J$. For response functions we will employ "probability of kill" which is computed as follows: Let $p^m_{i,j}$ be the probability that a single weapon of type i, allocated to aimpoint j, destroys target m, conditioned on the weapon's arrival at its destination. The probability that a type-i weapon arrives at its destination, its "reliability," is given by ρ_i. Because we have assumed independence of weapon effects, it is not difficult to compute the total probability that target m is destroyed by allocation z, which is

$$f_m(\underset{\sim}{z}) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} (1-\rho_i p^m_{i,j})^{z_{i,j}} .$$

Similarly, we denote by $p^n_{i,j}$ the conditional probability that a single type-i weapon allocated to aimpoint j destroys

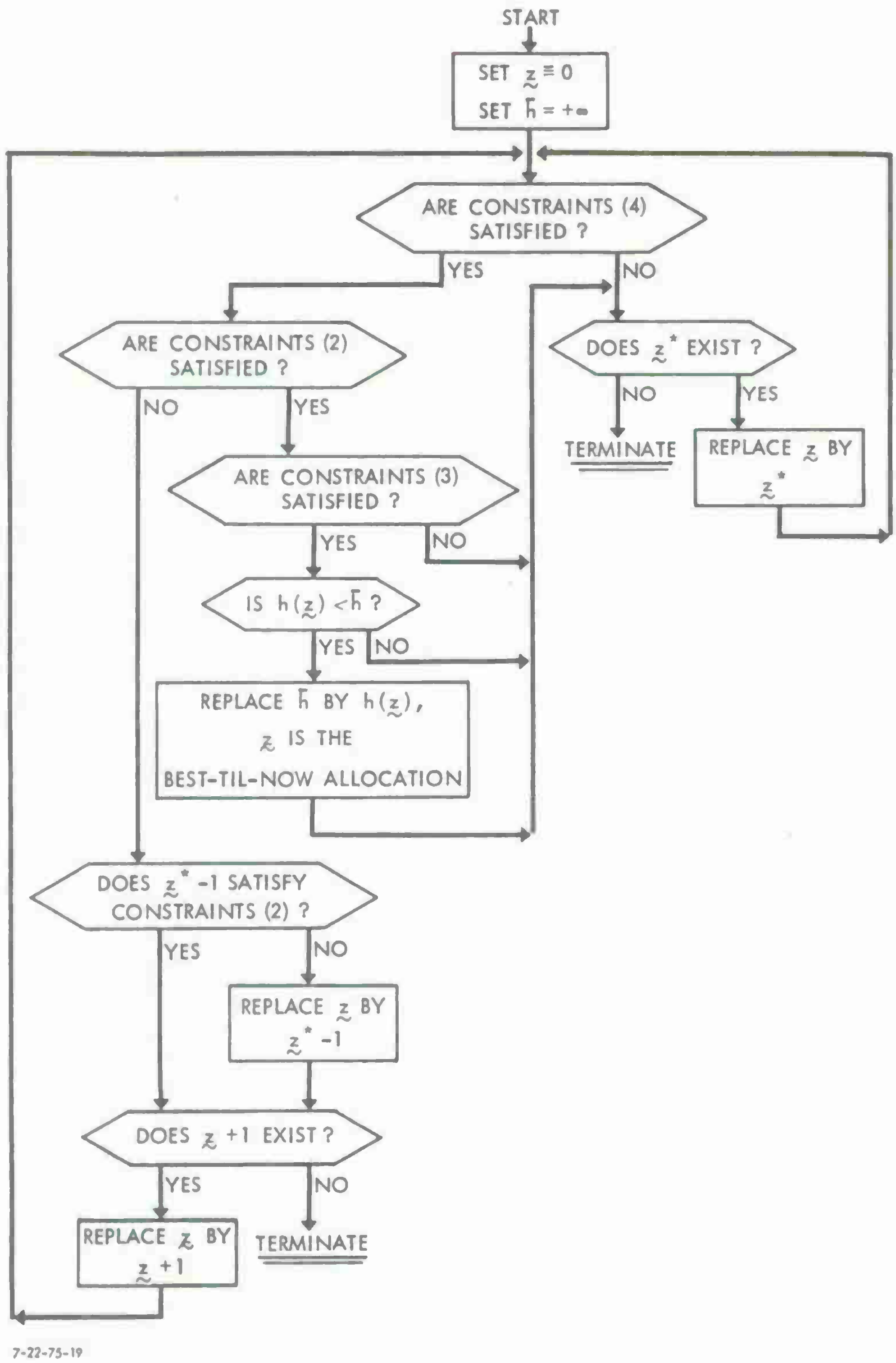

Figure 1. AN IMPLICIT ENUMERATION ALGORITHM

nontarget n. Therefore, the probability that allocation $\underset{\sim}{z}$ destroys nontarget n is

$$g_n(\underset{\sim}{z}) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} (1-\rho_i p_{i,j}^n)^{z_{i,j}} .$$

Although the values of the parameters $\{p_{i,j}^m\}$ and $\{p_{i,j}^n\}$ can be entirely arbitrary, within the obvious limits

$$0 \leq p_{i,j}^m \leq 1 \qquad m=1,\ldots,M;\ i=1,\ldots,I;\ j=1,\ldots,J ,$$

$$0 \leq p_{i,j}^n \leq 1 \qquad n=1,\ldots,N;\ i=1,\ldots,I;\ j=1,\ldots,J ,$$

we will use, for tutorial purposes, the following formulae, which are not unreasonable approximations to certain types of weapon damage curves and have been proposed by other analysts (see, for example, Eckler, Reference [1], or McNolty, Reference [4]):

$$\begin{aligned} p_{i,j}^m &= \exp\left\{-\alpha_{i,m}[(x_m-\xi_j)^2+(y_m-\zeta_j)^2]\right\} \quad m=1,\ldots,M;\ i=1,\ldots,I;\ j=1,\ldots,J \\ p_{i,j}^n &= \exp\left\{-\beta_{i,n}[(\mu_n-\xi_j)^2+(\nu_n-\zeta_j)^2]\right\} \quad n=1,\ldots,N;\ i=1,\ldots,I;\ j=1,\ldots,J \end{aligned} \tag{10}$$

where all $\alpha_{i,m}$, $\beta_{i,n}$ are nonnegative real numbers. The parameters $\{\alpha_{i,m}\}$ and $\{\beta_{i,n}\}$ are measures of the rate at which weapon effects decrease with distance.

With these conventions, we can now write explicitly the problem P′ which comprises this class of examples:

P′: Given nonnegative weights λ_n, n=1,...,N, and the values of

$$\begin{array}{ll} c_m \varepsilon [0,1] & m=1,\ldots,M \\ d_n \varepsilon [0,1] & n=1,\ldots,N \\ w_i \varepsilon \mathbb{Z}^+ & i=1,\ldots,I \\ \rho_i \varepsilon [0,1] & i=1,\ldots,I \end{array}$$

$$\alpha_{i,m} \geq 0 \qquad i=1,\ldots,I;\ m=1,\ldots,M$$

$$\beta_{i,n} \geq 0 \qquad i=1,\ldots,I;\ n=1,\ldots,N$$

$$x_m, y_m \qquad m=1,\ldots,M$$

$$\mu_n, \nu_n \qquad n=1,\ldots,N$$

$$\xi_j, \zeta_j \qquad j=1,\ldots,J$$

$$\underset{\underset{\sim}{z}}{\text{minimize}}\ h(\underset{\sim}{z}) = \sum_{n=1}^{N} \lambda_n \left\{ 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} \left(1 - \rho_i \exp\left\{ -\beta_{i,n}\left[(\mu_n - \xi_j)^2 + (\nu_n - \zeta_j)^2 \right] \right\} \right)^{z_{i,j}} \right\}$$

subject to

$$f_m(\underset{\sim}{z}) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} \left(1 - \rho_i \exp\left\{ -\alpha_{i,m}\left[(x_m - \xi_j)^2 + (y_m - \zeta_j)^2 \right] \right\} \right)^{z_{i,j}} \geq c_m \qquad m=1,\ldots,M$$

$$g_n(\underset{\sim}{z}) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} \left(1 - \rho_i \exp\left\{ -\beta_{i,n}\left[(\mu_n - \xi_j)^2 + (\nu_n - \zeta_j)^2 \right] \right\} \right)^{z_{i,j}} \leq d_n \qquad n=1,\ldots,N$$

$$\sum_{j=1}^{J} z_{i,j} \leq w_i \qquad i=1,\ldots,I$$

$$z_{i,j} \in \mathbb{Z}^+ \qquad i=1,\ldots,I;\ j=1,\ldots,J\ .$$

IV. COMPUTER APPLICATIONS

A FORTRAN routine to solve problems of the type given by P′ was written for the CDC 6400 computer, and was used to solve the numerical example of this section. (A listing of this program, along with input formats are given in the Appendix.) The values of the parameters are listed in Tables 1-6. The configuration of the targets, nontargets and aimpoints is depicted in Figure 2.

The routine ran for five seconds to compute the optimal solution, $\hat{\underset{\sim}{z}}$, given in Table 7.

Table 1. TARGET PARAMETERS

M=2

m	x_m	y_m	c_m
1	-1	0	.8
2	1	0	.8

Table 2. NONTARGET PARAMETERS

N=4

n	μ_n	ν_n	λ_n	d_n
1	-2	0	2	.3
2	-1	-1	4	.3
3	1	1	6	.3
4	2	0	8	.3

Table 3. AIMPOINT PARAMETERS

J=5

j	ξ_j	ζ_j
1	-1	1
2	-1	0
3	0	0
4	1	0
5	1	-1

Table 4. WEAPON PARAMETERS

I=2

i	w_i	p_i
1	6	.9
2	6	.7

Table 5. COMPONENTS OF $\underset{\sim}{\alpha}$

i \ m	1	2
1	.1	.1
2	.5	.5

Table 6. COMPONENTS OF $\underset{\sim}{\beta}$

i \ n	1	2	3	4
1	.05	.1	.1	.09
2	.8	.8	.8	.8

Table 7. OPTIMAL ALLOCATION $\hat{\underset{\sim}{z}}$

$\hat{\underset{\sim}{z}}$

i \ j	1	2	3	4	5
1	0	0	0	0	0
2	2	0	1	0	2

$h(\hat{\underset{\sim}{z}}) = 5.2$ $\quad g_1(\hat{\underset{\sim}{z}}) = .28$

$f_1(\hat{\underset{\sim}{z}}) = .83$ $\quad g_2(\hat{\underset{\sim}{z}}) = .24$

$f_2(\hat{\underset{\sim}{z}}) = .83$ $\quad g_3(\hat{\underset{\sim}{z}}) = .24$

$g_4(\hat{\underset{\sim}{z}}) = .28$

Table 8. OPTIMAL ALLOCATION $\underset{\sim}{z}'$

$\underset{\sim}{z}'$

i \ j	1	2	3	4	5
1	0	0	0	0	0
2	0	0	3	0	0

$h(\underset{\sim}{z}') = 4.5$ $\quad g_1(\underset{\sim}{z}') = .08$

$f_1(\underset{\sim}{z}') = .81$ $\quad g_2(\underset{\sim}{z}') = .37$

$f_2(\underset{\sim}{z}') = .82$ $\quad g_3(\underset{\sim}{z}') = .37$

$g_4(\underset{\sim}{z}') = .08$

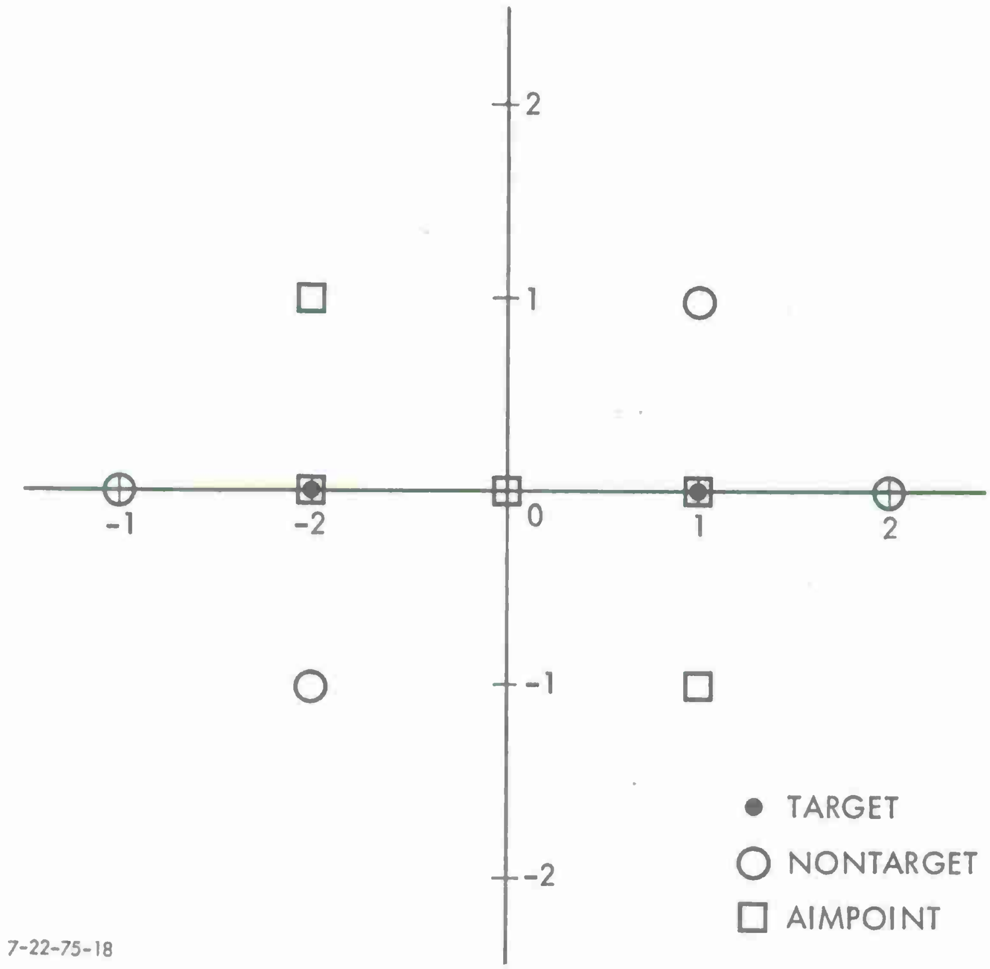

Figure 2. CONFIGURATION OF THE EXAMPLE

It is interesting to note that if all the d_i are changed to 1.0, which is equivalent to removing the individual nontarget damage constraints, then the optimal allocation is $\underset{\sim}{z}'$, given in Table 8. In this latter case, we have reduced total collateral damage over that given in Table 7, but only at the expense of considerably greater damage to two of the nontargets.

REFERENCES

[1] Eckler, A. R. and S. A. Burr. *Mathematical Models of Target Coverage and Missile Allocation*. Alexandria, Va.: Military Operations Research Society, 1972.

[2] Grotte, J. H. *A Nontarget Damage Minimizer (NDM) Which Achieves Targeting Objectives*, Paper P-1106 (also WSEG Report 256. Arlington, Va.: Weapons Systems Evaluation Group). Arlington, Va.: Institute for Defense Analyses, November 1975. (Vol. I, Unclassified; Vol. II, Secret).

[3] Lawler, E. L. and M.D. Bell. "A Method for Solving Discrete Optimization Problems." *Operations Research*, 14 (1966), 1098-1112.

[4] McNolty, F. "Expected Coverage for Targets of Nonuniform Density." *Operations Research*, 16 (1968), 1027-1040.

APPENDIX

FORTRAN LISTING AND INPUT SPECIFICATIONS

FORTRAN LISTING

```
      PROGRAM MDLTWO(INPUT,OUTPUT)
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      CALL READIT
      CALL CALCPRB
      CALL LEXO
      CALL OUT
      END
```

```
      SUBROUTINE READIT
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      READ1,NOTAR,NONON,NOAIM,NOWEAP
    1 FORMAT(4I10)
      DO 5 I=1,NOTAR
      READ10,XTAR(I),YTAR(I),DEST(I)
10    FORMAT(3F10.6)
5     CONTINUE
      DO 15I=1,NONON
      READ20,XNON(I),YNON(I),FACTOR(I),UPNOND(I)
20    FORMAT(4F10.6)
15    CONTINUE
      DO 25I=1,NOAIM
      READ30,XAIM(I),YAIM(I)
30    FORMAT(2F10.6)
25    CONTINUE
      DO100I=1,NOWEAP
      READ40,IWNUM(I),RELBL(I)
40    FORMAT(I10,F10.6)
      READ200,(EFFTAR(I,M),M=1,NOTAR)
      READ300,(EFFNON(I,N),N=1,NONON)
200   FORMAT(8F10.6/,2F10.6)
300   FORMAT(8F10.6/,2F10.6)
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALCPRB
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      DO10M=1,NOTAR
      DO10I=1,NOWEAP
      DO10J=1,NOAIM
      WW=EFFTAR(I,M)*((XAIM(J)-XTAR(M))**2+(YAIM(J)-YTAR(M))**2)
      PKT(M,I,J)=RELBL(I)*EXP(-WW)
10    CONTINUE
      DO20N=1,NONON
      DO20I=1,NOWEAP
      DO20J=1,NOAIM
      WW=EFFNON(I,N)*((XAIM(J)-XNON(N))**2+(YAIM(J)-YNON(N))**2)
      PKN(N,I,J)=RELBL(I)*EXP(-WW)
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LEXO
C     THIS SUBROUTINE FINDS AND STORES THE OPTIMAL ALLOCATION USING
C     LEXICOGRAPHIC ENUMERATION AFTER LAWLER-BELL. OPTIMAL VALUES ARE
C     STORED AS FOLLOWS--
C     BTNNTS--TOTAL NONTARGET SURVIVAL LEVEL  (FROM NTS)
C        =999999999. IF NO FEASIBLE SOLUTION IS FOUND
C     IBTNAL(.,.)--OPTIMAL ALLOCATION
C     BTNTS(.)--RESULTING TARGET SURVIVAL LEVEL (FROM TARS)
C     BTNNTV(.)--INDIVIDUAL NONTARGET SURVIVAL LEVELS  (FROM NTS)
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      REAL NTV
      INTEGER ITEMPAL(1000)
      INTEGER ITEMST(1000)
      DO 9000 LL=1,M
      ITEMST(LL)=0
 9000 CONTINUE
      BTNNTS=9999999999.
      M=NOWEAP*NOAIM
      DO 1 K=1,M
      IPRESAL(K)=0
    1 CONTINUE
C     BEGIN ENUMERATION
C     CHECK ZERO VECTOR FOR FEASIBILITY
      CALL TARS
      IF(IFLAG.NE.0)GO TO 100
C     IF HERE, NO ADDITIONAL WEAPONS ARE NEEDED
 8002 CALL NTS
      IF(NONFLAG.NE.0)RETURN
 8003 BTNNTS=SV
      DO 10 K=1,NONON
      BTNNTV(K)=NTV(K)
   10 CONTINUE
      DO 11 K=1,M
      IBTNAL(K)=IPRESAL(K)
   11 CONTINUE
      DO 12 K=1,NOTAR
      BTNTS(K)=TARSURV(K)
   12 CONTINUE
      RETURN
C     THIS SECTION COMPUTES NEXT ALLOCATION
  310 DO 315 J=1,NOAIM
      DO 315 I=1,NOWEAP
      KKK=I+(J-1)*NOWEAP
      IF(IPRESAL (KKK).LT.IWNUM(I))GO TO 320
      IPRESAL(KKK)=0
  315 CONTINUE
C     HERE IF IPRESAL WAS LAST ALLOCATION
      RETURN
  320 IPRESAL(KKK)=IPRESAL(KKK)+1
  399 CALL NUMBS
```

```
      IF(NFLAG.EQ.1)GO TO 600
  400 CALL TARS
      IF(IFLAG.NE.1)GO TO 500
C     HERE IF ALLOCATION INFEASIBLE
C     STORE IPRESAL
  100 DO 405 K=1,M
      ITEMPAL(K)=IPRESAL(K)
  405 CONTINUE
C     NOW TO COMPUTE IPRESALSTAR-1
      DO 410 K=1,M
      IF(IPRESAL(K).EQ.0)GO TO 410
      IPRESAL(K)=0
      GO TO 415
  410 CONTINUE
  415 IF(K.GE.M)GO TO 420
      L=K+1
      DO 425 K=L,M
      J=(K-1)/NOWEAP
      I=K-NOWEAP*J
      IF(IPRESAL(K).LT.IWNUM(I))GO TO 430
      IPRESAL(K)=0
  425 CONTINUE
      GO TO 420
  430 IPRESAL(K)=IPRESAL(K)+1
      GO TO 435
  420 DO 440 I=1,NOWEAP
      DO 440 J=1,NOAIM
      KKK=I+(J-1)*NOWEAP
      IPRESAL(KKK)=IWNUM(I)
  440 CONTINUE
      GO TO 480
  435 DO 445 J=1,NOAIM
      DO 445 I=1,NOWEAP
      KKK=I+(J-1)*NOWEAP
      IF(IPRESAL(KKK).NE.0)GO TO 450
      IPRESAL(KKK)=IWNUM(I)
  445 CONTINUE
  450 IPRESAL(KKK)=IPRESAL(KKK)-1
C     NOW WE HAVE IPRESALSTAR -1
  480 DO 9005 LL=1,M
      IF(ITEMSI(LL).NE.IPRESAL(LL))GO TO 9010
 9005 CONTINUE
      GO TO 9020
 9010 DO 9015 LL=1,M
      ITEMST(LL)=IPRESAL(LL)
 9015 CONTINUE
      CALL TARS
      IF(IFLAG.NE.0)GO TO 310
C     HERE IF IPRESALSTAR -1 IS FEASIBLE
 9020 CONTINUE
      DO 495 K=1,M
      IPRESAL(K)=ITEMPAL(K)
  495 CONTINUE
      GO TO 310
  500 CALL NTS
      IF(NONFLAG.NE.0)GOTO600
 8010 IF(SV.GE.BTNNTS)GO TO 600
C     HAVE FOUND A NEW OPTIMUM
```

```
      DO 510 K=1,M
      IBTNAL(K)=IPRESAL(K)
  510 CONTINUE
      DO 515 I=1,NOTAR
      BTNTS(I)=TARSURV(I)
  515 CONTINUE
      BTNNTS=SV
      DO 520 I=1,NONON
      BTNNTV(I)=NTV(I)
  520 CONTINUE
C     SKIP TO IPRESALSTAR
  600 DO 610 K=1,M
      IF(IPRESAL(K).EQ.0)GO TO 610
      IPRESAL(K)=0
      GO TO 620
  610 CONTINUE
  620 IF(K.GE.M) RETURN
      L=K+1
      DO 625  K=L,M
      J=(K-1)/NOWEAP
      I=K-NOWEAP*J
      IF(IPRESAL(K).LT.IWNUM(I))GO TO 630
      IPRESAL(K)=0
  625 CONTINUE
      RETURN
  630 IPRESAL(K)=IPRESAL(K)+1
      GO TO 399
      END
```

```
      SUBROUTINE TARS
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      DO10M=1,NOTAR
      TSURV=1.
      DO50I=1,NOWEAP
      DO50J=1,NOAIM
      KKK=I+(J-1)*NOWEAP
      IF(IPRESAL(KKK).LE.0)GOTO50
      IF(PKT(M,I,J).GE.1.)8,9
 8    TSURV=0.
      GO TO 50
 9    PS=(1.-PKT(M,I,J))**IPRESAL(KKK)
      TSURV=TSURV*PS
 50   CONTINUE
C     ARE CONSTRAINTS SATISFIED
      WW=1.-DEST(M)
      IF(TSURV.GT.WW)6,7
 6    IFLAG=1
      RETURN
 7    TARSURV(M)=1.-TSURV
 10   CONTINUE
      IFLAG=0
      RETURN
      END
```

```
      SUBROUTINE NTS
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      REAL NTV
      SV=0.
      DO1000N=1,NONON
      TEMPSV=1.
      DO50I=1,NOWEAP
      DO50J=1,NOAIM
      KKK=I+(J-1)*NOWEAP
      IF(IPRESAL(KKK).LE.0)GOTO50
      IF(PKN(N,I,J).GE.1.)8,9
8     TEMPSV=0.
      GO TO 50
9     PS=(1.-PKN(N,I,J))**IPRESAL(KKK)
      TEMPSV=TEMPSV*PS
50    CONTINUE
      WW=1.-TEMPSV
      IF(WW.GT.UPNOND(N))7,10
7     NONFLAG=1
      RETURN
10    SV=SV+FACTOR(N)*WW
      NTV(N)=FACTOR(N)*WW
1000  CONTINUE
      NONFLAG=0
      RETURN
      END
```

```
      SUBROUTINE NUMBS
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      DO1I=1,NOWEAP
      ISUM=0
      DO2J=1,NOAIM
      KKK=I+(J-1)*NOWEAP
      ISUM=IPRESAL(KKK)+ISUM
2     CONTINUE
      IF(ISUM.LE.IWNUM(I))GOTO1
      NFLAG=1
      RETURN
1     CONTINUE
      NFLAG=0
      RETURN
      END
```

```
      SUBROUTINE OUT
      COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
      COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
      COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
       COMMON/AIMPNTS/XAIM(100),YAIM(100)
      COMMON/EWEP/IWNUM(10),RELBL(10),EFFTAR(10,10),EFFNON(10,10)
      COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),
     1TARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
     2,BTNNTV(10),NONFLAG
      REAL NTV
      PRINT5
5     FORMAT(1H1,*PROGRAM MDL*)
      PRINT10
10    FORMAT(1H-,*INPUT DATA*)
      PRINT15,NOTAR
15    FORMAT(1H-,*TARGETS (*,I4,*)*)
      PRINT20
20    FORMAT(1H0,*TARGET NUMBER*,6X,*X COORD.*,7X,*Y COORD.*,6X,
     1*PROB. OF DEST.*/)
      DO26I=1,NOTAR
      PRINT25,I,XTAR(I),YTAR(I),DEST(I)
25    FORMAT(1H ,7X,I2,10X,F10.3,5X,F10.3,5X,F10.3)
26    CONTINUE
      PRINT111
111   FORMAT(1H-)
      PRINT30,NONON
30    FORMAT(1H-,*NONTARGETS (*,I4,*)*)
      PRINT35
35    FORMAT(1H0,*NONTAR NUMBER*,6X,*X COORD.*,7X,*Y COORD.*,12X,
     1*VALUE*, 7X,*DAMAGE LIMIT*/)
      DO41I=1,NONON
      PRINT36,I,XNON(I),YNON(I),FACTOR(I),UPNOND(I)
36    FORMAT(1H ,7X,I2,10X,F10.3,5X,F10.3,5X,F10.3,5X,F10.3)
41    CONTINUE
      PRINT111
      PRINT45,NOAIM
45    FORMAT(1H-,*AIMPOINTS (*,I4*)*)
      PRINT50
50    FORMAT(1H0,*AIMPNT NUMBER*,6X,*X COORD.*,7X,*Y COORD.*/)
      DO56I=1,NOAIM
      PRINT55,I,XAIM(I),YAIM(I)
55    FORMAT(1H ,6X,I3,10X,F10.3,5X,F10.3)
56    CONTINUE
      PRINT111
      PRINT60,NOWEAP
60    FORMAT(1H-,*WEAPON CLASSES (*,I4,*)*)
      PRINT65
65    FORMAT(1H0,*CLASS NUMBER*,5X,*TOTAL AVAILABLE*,5X,*RELIABILITY*,
     1/)
      DO71I=1,NOWEAP
      PRINT70,I,IWNUM(I),RELBL(I)
70    FORMAT(1H ,5X,I2,15X,I4,15X,F10.3)
71    CONTINUE
      PRINT100
100   FORMAT(1H-,*WEAPON-TARGET EFFECTIVENESS TABLE*)
      PRINT 101
101   FORMAT(1H-,*      /TARGET*)
```

```
      PRINT102
102   FORMAT(1H ,*WEAPON/*)
      PRINT103,(I,I=1,NOTAR)
103   FORMAT(1H0,14X,I2,9(8X,I2)/)
      DO105I=1,NOWEAP
      PRINT110,I,(EFFTAR(I,J),J=1,NOTAR)
110   FORMAT(1H ,4X,I2,4X,10F10.3)
105   CONTINUE
      PRINT111
      PRINT200
200   FORMAT(1H-,*WEAPON-NONTARGET EFFECTIVENESS TABLE*)
      PRINT201
201   FORMAT(1H-,*      /NONTARGET*)
      PRINT 102
      PRINT203,(I,I=1,NONON)
203   FORMAT(1H0,14X,I2,9(8X,I2)/)
      DO205I=1,NOWEAP
      PRINT210,I,(EFFNON(I,J),J=1,NONON)
210   FORMAT(1H ,4X,I2,4X,10F10.3)
205   CONTINUE
      PRINT111
      PRINT111
      PRINT1005
1005  FORMAT(1H-,*ALLOCATION RESULTS*)
      IF(BTNNTS.LT.9999999999.)GOTO1100
      PRINT1010
1010  FORMAT(1H0,*IT IS IMPOSSIBLE TO MEET THE TARGET DAMAGE CONSTRAINTS
     1--PROBLEM INFEASIBLE*)
      RETURN
1100  PRINT 1125
1125  FORMAT(1H0,*FOLLOWING IS THE OPTIMAL ALLOCATION*)
      PRINT1130
1130  FORMAT(1H0,*WEAPON CLASS*,5X,*AIMPOINT*,5X,*NUMBER ASSIGNED*/)
      DO1136I=1,NOWEAP
      DO1136J=1,NOAIM
      KKK=I+(J-1)*NOWEAP
      IF(IBTNAL(KKK).LE.0)GOTO1136
      PRINT1135,I,J,IBTNAL(KKK)
1135  FORMAT(1H ,5X,I2,13X,I3,10X,I10)
1136  CONTINUE
      PRINT111
      PRINT1205
1205  FORMAT(1H-,*TARGET DAMAGE*)
      PRINT1210
1210  FORMAT(1H-,*TARGET CLASS*,5X,*RESULTING PROB. OF DEST.*,5X,*SPECIF
     1IED PROB. OF DEST.*/)
      DO 1216I=1,NOTAR
      PRINT 1215,I,BTNTS(I),DEST(I)
1215  FORMAT(1H ,2X,I2,14X,F10.3,17X,F10.3)
1216  CONTINUE
      PRINT111
      TOT=0.
      DO1220I=1,NONON
      TOT=TOT+FACTOR(I)
1220  CONTINUE
      PRINT1225,TOT
1225  FORMAT(1H0,*ORIGINAL TOTAL NONTARGET VALUE WAS *,F10.3)
      PC=BTNNTS/TOT*100.
```

```
      PRINT1230,BTNNTS,PC
 1230 FORMAT(1H-,*TOTAL EXPECTED NONTARGET VALUE DESTROYED IS *,F10.3,
     1*   OR   *,F10.3,*  PERCENT.*)
      PRINT111
      PRINT1235
 1235 FORMAT(1H-,*INDIVIDUAL NONTARGET EXPECTED VALUE DESTROYED LISTED B
     1ELOW*)
      PRINT1240
 1240 FORMAT(1H-,*NONTARGET NUMBER*,5X,*ORIGINAL VALUE*,5X,
     1*EXPECTED VALUE DESTROYED*,10X,*PERCENT*,5X,*SPECIFIED MAXIMUM (PE
     2RCENT)*/)
      DO 1246I=1,NONON
      WWW=100.*UPNOND(I)
      PCC=BTNNTV(I)/FACTOR(I)*100.
      PRINT1245,I,FACTOR(I),BTNNTV(I),PCC,WWW
 1245 FORMAT(1H ,7X,I2,14X,F10.3,14X,F10.3,15X,F10.3,10X,F10.3)
 1246 CONTINUE
      RETURN
      END
```

INPUT SPECIFICATIONS

Refer to problem P´ for notation.

Core requirements impose the following limits:

$M \leq 10$

$N \leq 10$

$J \leq 100$

$I \leq 10$.

Card Name	Input Parameters	Format
LIMITS	M, N, J, I	4I10
TARGET (1 card each)	x_m, y_m, c_m	3F10.6
NONTARGET (1 card each)	μ_n, ν_n, λ_n, d_n	4F10.6
AIMPOINT (1 card each)	ξ_j, ζ_j	2F10.6
WEAPON (1 deck each)	w_i, ρ_i	I10,F10.6
	$\alpha_{i,1}$, $\alpha_{i,2}$, ...	8F10.6/2F10.6
	$\beta_{i,1}$, $\beta_{i,2}$, ...	8F10.6/2F10.6